

Ontologías en Arquitecturas Dirigidas por Modelos

López, Gustavo (1); Servetto, Arturo (2); Echeverría, Adriana (1); Jeder, Ismael (1); Grossi, María (2); Jiménez Rey, Elizabeth (2)

(1) Laboratorio de Informática de Gestión y (2) Laboratorio de Sistemas Operativos y Bases de Datos

Departamento de Computación, Facultad de Ingeniería, UBA

Av. Paseo Colón 850 - (C1063ACV) Ciudad de Buenos Aires

TE: 4343-0893/0092

glopez@fi.uba.ar, aserve@gmail.com

Resumen

La arquitectura dirigida por modelos (MDA) y las ontologías son dos de los recursos cada vez más populares dentro de la comunidad informática para el desarrollo de sistemas de software. La MDA presenta un marco de trabajo para crear aplicaciones informáticas. Paralelamente, las ontologías devienen en recursos cuya función es facilitar la interacción entre herramientas de software heterogéneas. Los objetivos de ambos recursos, si bien son distintos, no son mutuamente excluyentes. Esto permite establecer una relación entre ellos. Este proyecto investiga, clasifica y utiliza los puntos de conexión entre MDA y ontologías, a partir del análisis de trabajos desarrollados en este campo, a fin de proponer una metodología para el desarrollo de Arquitecturas dirigidas por Modelos sustentada en el uso de ontologías.

Palabras clave: ontologías, MDA

Contexto

Proyecto de innovación tecnológica acreditado por la Secretaría de Ciencia y Técnica de la Universidad de Buenos Aires para la Programación Científica 2011-2014.

Introducción

Como resultado de las mejores prácticas en Ingeniería de Software surge la Arquitectura dirigida por Modelos (MDA). Este tipo de arquitectura propone un marco de trabajo basado en el uso de modelos para el desarrollo de software [3] [4]. Los modelos desarrollados pueden clasificarse de acuerdo al punto de vista del sistema que muestren y están relacionados entre sí a través de transformaciones que permiten pasar de un modelo a otro.

Las Ontologías han surgido como una manera de estandarizar y facilitar la interacción de los sistemas actuales, como una manera de representar el conocimiento compartido acerca de un dominio determinado [5]. Por lo general, las ontologías están asociadas a la idea de contenedores semánticos [6], esto es a través de vocabularios controlados o bien por estructuras que soportan todo el peso semántico de una interacción entre servicios: tienen algún lugar dentro de la arquitectura de los sistemas [26].

Las ontologías se implementan en el diseño de aplicaciones mediante el uso de artefactos de software. Siendo un artefacto, las ontologías pueden ser construidas de manera ad hoc o se puede seguir un estándar de construcción de aplicaciones [20].

No hay diferencias sustanciales entre el modelado y la representación de ontologías [14]. De este modo, la solución debe ser una unificación de ambas técnicas que tome de las onto-

logías la representación formal a través del uso de Lógica Descriptiva que logre una semántica formal consistente y de MDA los mecanismos de representación a través de sus múltiples niveles, metaniveles y dimensiones [18] [24].

Las ontologías facilitan la reutilización y generan confiabilidad en las aplicaciones, dado que permiten sistematizar el chequeo de la consistencia [7] [12]. También mejoran la documentación del software y en consecuencia reducen los costos de mantenimiento [29].

Parrend y David [12] presentan un proceso basado en ontologías del dominio para apoyar los procesos de ingeniería basada en modelos. Algo similar propone Pahl [13] para el diseño de servicios web. Otros usos de las ontologías existentes se enfocan en la recopilación de requisitos de diferentes dominios como los modelos de procesos [31] y los componentes de software [15]. Además, Chitchyan y otros [16] las emplean en el diseño de aplicaciones orientadas a aspectos y Ferreiro y otros [17] en la construcción de bases de datos desde documentos de la Web.

Devedzic [18] propone la construcción de ontologías a partir de patrones de diseño y Romain y Cuesta [19] proponen un enfoque basado en aspectos para la construcción de ontologías del dominio durante el desarrollo de sistemas de información.

La generación automática de código, como una solución para cerrar la brecha existente entre las etapas de análisis y diseño a la implementación, es fuente de muchos trabajos de investigación. Las ontologías se usan, en este contexto, de diversas formas.

Bures y otros [20] definen la síntesis de programa (*program synthesis*) como el proceso para derivar, automáticamente, código ejecutable desde especificaciones de alto nivel no ejecutables.

Bench-Capon [21] utiliza ontologías para verificar la coherencia de una base de conocimientos, proveer un medio para estructurar las pruebas y sugerir respuestas apropiadas cuando las pruebas indican que existen fallas. Looker y otros [22] proponen un método para medir la

capacidad de un sistema para ejecutar su función de una manera confiable, en el contexto de los servicios Web y las arquitecturas orientadas a servicios. Se construyeron varias ontologías: descripción de los servicios Web, una extensión de un modelo general de fallas y modos de errores. Yu y otros [23] proponen un método para probar la capacidad que poseen los servicios Web para operar entre ellos. El uso de la ontología permite detectar nuevos tipos de errores, permitiendo el ingreso de las reglas correspondientes.

La ontología de Kitchenham y otros [24] contiene los conceptos relevantes para la clasificación de estudios empíricos en el mantenimiento de software. Algunos tipos de conocimiento, implícitos en los artefactos construidos en el ciclo de vida de desarrollo de una aplicación, se representan como conceptos en la ontología presentada por Deridder [26]. Entre estos conocimientos, se encuentran las conexiones entre los diferentes artefactos, el conocimiento que se pierde en los refinamientos interactivos y el conocimiento considerado como de sentido común por las partes participantes en el desarrollo. Hyland-Wood y otros [27] construyen una ontología de conceptos de la Ingeniería de Software. Dichos conceptos corresponden a las componentes y metadatos del sistema y permiten la navegación sobre éste facilitando el entendimiento del software y su mantenimiento. April [28] formaliza una ontología de mantenimiento con base en la ontología de Ruiz y otros [29] y establece una relación con un conjunto de mejores prácticas contenidas en la integración de modelos de capacidad y madurez (CMMI).

La multiplicidad de esquemas generada en el proceso de síntesis de programas, se mejora con la introducción de una ontología de conceptos de dominios específicos, que guía la actualización y el cambio de los sistemas, actividades representativas de la fase de mantenimiento [31]. Nistor [30] se ocupa de hacer extensible la síntesis de sistemas, por medio de la ontología, a nuevos dominios de aplicación.

Líneas de investigación y desarrollo

Generación de modelos con base en Ontologías

En esta línea se consideran desarrollos cuyo punto de partida es una ontología, la cual a través de transformaciones es convertida en un modelo. Ejemplos de este grupo son los trabajos de Wongthongtham [4], Johansson [9] y Soares [10]. La característica principal de esta línea es la tendencia a la reutilización de ontologías y uno de los principales motivos para el uso de una ontología como base de modelado es que éstas representan el conocimiento concertado sobre un dominio determinado, generalmente dado por una comunidad especializada en dicho dominio. De este modo, los modelos obtenidos por esta técnica pueden ser más completos en pocas interacciones comparados con los obtenidos por un método ad hoc. También la ontología constituye el punto de referencia sobre el cual es comparado el modelo final para ver si cumple o no con los requerimientos y la conceptualización inicial [31].

Wongthongtham, de Palol y Olivé [4], proponen la creación de Modelos Conceptuales partiendo de una Ontología. Este método propone obtener una ontología del dominio sobre el cual se desea hacer un desarrollo. En este paso se puede crear la ontología o usar una ya existente que se encuentre disponible [7]. La ontología permitirá dar una visión más amplia para la especificación del sistema y reducirá el número de aspectos no contemplados en el desarrollo. A través de procesos de refinamiento de la ontología y de validación del modelo contra la ontología, se obtendrá un modelo conceptual de mejor calidad para que sea el soporte de desarrollo de cualquier aplicación. Los procesos de refinamiento y validación, realizan el papel de modelos de transformación que usa MDA.

Un caso similar es presentado por Soares, Heckel y Lohmann [10] quienes se basan en una ontología para la construcción de modelos en el desarrollo de Servicios Web. La ontología nue-

vamente es tomada como base y la generación de los modelos es realizada a través de la aplicación de "Reglas de Transformación de Grafos". Johansson, Pan, Oberle, Wallace, Uschold y Kendel [9], presentan las ontologías como Modelos Formales de Especificación. Siendo las ontologías instrumentos para facilitar el entendimiento entre diversos agentes tomando en cuenta las propiedades semánticas de la comunicación y siendo reflejadas estas propiedades en un entorno de formalismos lógicos, estas se convierten en muy buenos instrumentos no solo para la creación de modelos conceptuales sino también para la creación de Modelos de Datos que dan completitud a la aplicación de la que son soporte [31].

Generación de Ontologías con base en modelos

Esta línea abarca los desarrollos en los cuales las ontologías son el producto de procesos de transformación de uno o varios modelos. Las ontologías obtenidas son el reflejo de la conceptualización sobre la cual se han basado uno o varios desarrollos. Astrova y Stantic [1] nos muestran un caso de ingeniería inversa, en el cual basado en un Modelo de Datos podemos llegar a la Ontología.

Pahl [13] resalta la importancia de tener un vocabulario común que sirva de soporte a todos los elementos involucrados en un desarrollo y que sea utilizado durante todo el proceso de desarrollo de la aplicación. Asimismo, resalta cómo una ontología podría compilar el punto de vista utilizado por una arquitectura. De este modo, si un punto de vista describe un sistema desde una única perspectiva, se podría concluir que varias aplicaciones desarrolladas bajo el mismo punto de vista y centradas en el mismo dominio deben tener una ontología común. Este punto es la formalización de la tarea de recopilación de mejores experiencias para una tarea específica: modelar un dominio a partir de un punto de vista específico y plasmarlo en una ontología.

Astrova [1] evidencia que la relación puede ser 1 a 1, donde un modelo genera una ontolog-

ía. Sin embargo, Pahl [13] presenta una relación 1 a varios, donde varios puntos de vista sobre un problema generan una sola ontología.

Resultados y Objetivos

1. Definir un marco teórico/práctico que de sustento al uso de Ontologías en el marco de la construcción de Arquitecturas dirigidas por modelos.
 - 1.1. Presentar una visión unificada sobre aspectos y nomenclatura de la Arquitectura dirigida por modelos.
 - 1.2. Plantear un marco teórico/práctico general que cubra los aspectos del uso de Ontologías colaborando para la definición de Arquitecturas dirigidas por Modelos.
2. Especificar, desarrollar y evaluar un protocolo para la aplicación de Ontologías en relación a la construcción de Arquitecturas dirigidas por Modelos.
 - 2.1. Presentar una visión unificada sobre componentes de metodologías para el diseño de aplicaciones con Arquitectura dirigida por Modelos.
 - 2.2. Presentar una extensión del componente metodológico específico que custodie los aspectos correspondientes a la Arquitectura dirigida por Modelos (MDA) sustentada por la aplicación de Ontologías.

Formación de Recursos Humanos

El equipo de trabajo se compone de los autores, y actualmente se encuentra en desarrollo una tesis de grado de Ingeniería en Informática y un trabajo para la Licenciatura en Análisis de Sistemas

Referencias

[1] I. Astrova y B. Stantic, "Extracting Ontologies for the Semantic Web from HTML Forms" en Workshop

on Ontology and Enterprise Modelling: Indefinites for Interoperability. Vienna, Austria, 2004.

[2] N. Noy y D. McGuinness, "A Guide to Create your First Ontology. Stanford University",

<http://www.ksl.Stanford.edu/people/dlm/papers/ontology-tutorial-noymcguinness.htm> 2001. 44 10/24/02 10/24/02

[3] J. Miller y J. Mukerji, "OMG. OMG Model Driven Architecture (MDA) Guide". (eds) 2003. document number ormsc/2001-07-01.

Extractado en Mayo de 2005 de <http://www.omg.org/docs/omg/03-06-01.pdf>

[4] P. Wongthongtham y otros, "Ontology-based multi-site software development methodology and tools," *Journal of Systems Architecture*, vol. 52, no. 11, pp. 640-653, 2006.

[5] J. T. E. Timm y G. C. Gannod, "A Modeldriven Approach for Specifying Semantic Web Services". *Proceedings of the 3rd IEEE International Conference on Web Services (ICWS 2005)*, Julio 2005.

[6] M. Shamsfard y A. Abdollahzadeh Barforoush, "The state of the art in ontology learning: a framework for comparison". *The Knowledge Engineering Review*, Vol 18:4, p. 293-316. Cambridge University Press.

[7] M. Uschold, y R. Jasper, "A Framework for Understanding and Classifying Ontology Applications," en *Proceedings of the IJCAI Workshop on Ontologies and Problem-Solving Methods (KRR5)*, Stockholm, Sweden, 1999.

[8] J. Rech, y K.-D. Althoff, "Artificial Intelligence and software Engineering: Status and Future Trends," *Journal KI*, vol. 18, no. 3, pp. 5-11, 2004.

[9] R. Johansson y otros, "Carsim: A System to Visualize Written Road Accident Reports as Animated 3D Scenes," en *Proceedings of the ACL second Workshop on Text Meaning and Interpretation*, Barcelona, España, 2004, pp. 57-64.

[10] A. Soares, "A social and organisational ontological foundation for Enterprise Modelling," en *Proceedings of the IFAC Symposium on Manufacturing Modelling, Management and Control (MIM)*, Patras, Grecia, 2000.

[11] T. Yamaguchi, "Modeling software Processes by using Process and Object Ontologies," en *Proceed-*

ings of the 12th IEEE International Conference on Automated software Engineering, Incline Village, 1997, pp. 319-320.

[12] P. Parrend, y B. David, "Use of Ontologies as a Way to Automate MDE Processes," en Proc. of the IEEE EuroCon Conference, Belgrade, Serbia, 2005, pp. 567-570

[13] C. Pahl, "Semantic Model-Driven Architecting of Service-Based software Systems," *Information and software Technology* vol. 19, no. 8, pp. 838-850, 2007.

[14] M. Gnatz y otros, "Towards a Living software Development Process based on Process Patterns," *Lecture Notes in Computer Science*, vol. 2077, pp. 182-202, 2001.

[15] H. Happel y otros, "KOntoR: An Ontology-enabled Approach to software Reuse," en Proceedings of the 18th Intl. Conf. on software Engineering & Knowledge Engineering, San Francisco, 2006, pp. 349-354.

[16] R. Chitchyan y otros, *Initial Version of Aspect-Oriented Requirements Engineering Model*, Deliverable D36, Document No. AOSD-Europe-ULANC-17, AOSD-Europe, University of Lancaster, 2006.

[17] J. Ferreiro y otros, "Generación Automática de una Base de Datos desde Documentos de la Web," en Memorias del Congreso Argentino de Ciencias de la Computación, Ushuaia, 2000.

[18] V. Devedzic, "Ontologies: Borrowing from software Patterns," *Intelligence*, vol. 10, no. 3, pp. 14-24, 1999.

[19] M. Romay, y C. Cuesta, "Hacia la definición de Ontologías Orientadas a Aspectos," en Memorias del Taller de Desarrollo de software Orientado a Aspectos (DSOA), Granada, España, 2005.

[20] T. Bures y otros, "The role of ontologies in schema-based program synthesis," en Proceedings 19th annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA). Workshop on Ontologies as Software Engineering Artifacts, 2004.

[21] T. J. M. Bench-Capon, "The Role of Ontologies in the Verification and Validation of Knowledge Based Systems," en 9th International Workshop on

Database and Expert Systems Applications (DEXA), 1998.

[22] N. Looker y otros, "An Ontology-Based Approach for Determining the Dependability of Service-Oriented Architectures," en Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS), 2005, pp. 171-178.

[23] Y. Yu y otros, "Web services interoperability testing based on ontology," en Proceedings of the 5th International Conference on Computer and Information Technology (CIT), Shanghai, China, 2005, pp. 1075-1079.

[24] B. A. Kitchenham y otros, "Towards an ontology of software maintenance," *Journal of software Maintenance: Research and Practice* vol. 11, no. 6, pp. 365-389, 1999.

[25] K. M. Oliveira y otros, "Knowledge for software Maintenance," en Proceedings Fifteenth International Conference on software Engineering and Knowledge Engineering (SEKE), 2003, pp. 61-68.

[26] D. Deridder, *Facilitating software maintenance and reuse activities with a concept-oriented approach*, Brussels: Programming Technology Lab, Vrije Universiteit Brussel, 2002.

[27] D. Hyland-Wood y otros, "Enhancing software Maintenance by using Semantic Web Techniques," en International Semantic Web Conference (ISWC), Athens, GA, USA, 2006.

[28] A. April y otros, "A Formalism of ontology to support a software maintenance knowledge-based system," en Proceedings of the Eighteenth International Conference on software Engineering & Knowledge Engineering Conference (SEKE06), San Francisco, CA, USA, 2006, pp. 331-336.

[29] F. Ruiz y otros, "An Ontology For The Management Of software Maintenance Projects," *International Journal of software Engineering and Knowledge Engineering*, vol. 14, no. 3, pp. 323-349, 2004.

[30] E. Nistor, *Using Domain Models in Extensible Schema-based software Synthesis*; <http://www.ics.uci.edu/enistor/research/nistor-report.pdf>, California, 2004.

[31] A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams," en Proceedings of 29th International Conference on software Engineering. ICSE, Minneapolis, USA, 2007, pp. 85-103.